DOCUMENT RESUME

ED 082 501 EM 011 472

AUTHOR Bailey, F. N.; Kain, R. Y.
TITLE Project ITCH: Interactive Digital Simulation in Electrical Engineering Education.
INSTITUTION Minnesota Univ., Minneapolis. Dept. of Electrical Engineering.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE Jun 73
NOTE 13p.; Paper presented at the Conference on Computers in the Undergraduate Curricula (4th, Claremont, California, June 18-20, 1973)

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Assisted Instruction; Computer Graphics; Digital Computers; *Engineering Education; Higher Education; Input Output; Interaction; *Man Machine Systems; *Pilot Projects; Program Descriptions; *Simulation; Time Sharing; Undergraduate Study
IDENTIFIERS Bode Nyquist; CAI; Electrical Engineering; Fourier; Natural Language; *Project ITCH

ABSTRACT

A two-stage project is investigating the educational potential of a low-cost time-sharing system used as a simulation tool in Electrical Engineering (EE) education. Phase I involves a pilot study and Phase II a full integration. The system employs interactive computer simulation to teach engineering concepts which are not well handled by traditional methods. It requires less mathematical sophistication of the student and provides him with a measure of control, but has the drawbacks of being less flexible and realistic than ordinary laboratory experiments. The ITCH approach provides graphic input and output and employs natural language dialogs. The hardware is based upon the Digital Equipment Corporation PDP-8 line. Phase I software developments include the modification of a TSS/8 monitor and the design of instructional packages dealing with antennae Bode/Nyquist problems, point/line charges, Fourier problems, static fields transistors, root loci, phases and planes, linear filters, and amplifier design. Each package consists of a problem with appropriate inputs and outputs. Students specify parameters and request displays of results, thereby gaining insights into man-machine interactions and parameter-process relations. The packages developed to date are being tested with EE faculty and third year students. (LB)

FILMED FROM BEST AVAILABLE COPY

PROJECT ITCH: INTERACTIVE DIGITAL SIMULATION IN ELECTRICAL ENGINEERING EDUCATION*

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

F. N. Bailey and R. Y. Kain
Department of Electrical Engineering
University of Minnesota
Minneapolis, Minnesota 55455
(612) 373-2851

ED 082501

Project ITCH is an experiment to investigate the educational potential of a low cost time-sharing system used as a simulation tool in Electrical Engineering education. The simulations are pre-programmed so the student need not be a programmer to use the system. Conceptually, Project ITCH may be viewed as an advanced form of CAI although it goes well beyond most extant CAI systems in the sophistication of the concepts illustrated and the type of interaction provided. Many of the roots of this approach lie more in the philosophy of "hands-on simulation" which grew out of analog computing.

The ITCH experiment is planned in two phases: Phase I is a pilot study, demonstration phase and Phase II is a full scale system and integration phase. The phase goals are:

Phase I Goals

1. Feasibility study and concept demonstration on a small pilot system
2. Early involvement of a number of EE faculty members.

Phase II Goals

1. Expansion to a 16 terminal system
2. Integration of the developed tool into the EE curriculum

Phase I is supported by a two year grant from the National Science Foundation. At the time of this writing, we have about one year to go on Phase I.

In this paper we will outline the philosophy behind Project ITCH, describe Phase I ITCH hardware and software, and show some typical user-machine dialogs.

The ITCH Philosophy

In studying a complex process one obtains insight into the workings of the process by some combination of (a) analysis and (b) observation and experimentation. As a person begins to understand the process in question, he builds conceptual models (or simply concepts) to explain the observationally and/or analytically obtained data. While concept formation is important in any discipline, it has a much greater significance in the natural sciences where important core concepts must be assimilated as the building blocks for future work.

In an educational environment the insights necessary for concept formation are obtained through lectures, recitations, demonstrations, laboratories, and self-study. In the learning process the student attempts to form concepts by integrating the insight obtained through all of these experiences. While this approach is very effective in some cases, the dependence on mathematical models and explanations demands considerable mathematical sophistication on the part of the students in most engineering lecture courses. Thus we have long sequences of mathematical prerequisites to all Electrical Engineering course offerings. Yet it appears that many students can understand the mathematics only after they understand the engineering. For others the lecture approach is simply too abstract.

To overcome excessive abstraction we currently provide laboratories where the student encounters concrete examples of some of the abstract concepts covered in lectures. This mixed approach can be very successful, but unfortunately many important and difficult concepts cannot be illustrated in reasonable laboratory experiments. The primary importance of interactive computer simulation lies in its ability to complement both lectures and laboratories.

EM011 472

When compared with lectures, computer experiments have the advantages of requiring less mathematical sophistication and offering more opportunity for individual control of the

learning environment. Within the specific simulation framework the student is free to ask questions at a level and a rate set by his individual needs. At the same time, computer experiments are not constrained by the same cost, equipment and technical limitations as traditional laboratories. However, computer experiments are neither as flexible as lectures (due to the extensive software developments required) nor as realistic as laboratory experiments. Thus computer experiments offer a third basic alternative in the educational repertoire. A primary aim of Project ITCH is the exploration of ways in which interactive computer simulation can complement (rather than displace) existing approaches to concept formation [1].

Comment

Educational institutions have recently come under attack because of alleged rigidity and failure to adapt to individual student needs [2], [3]. Recent studies suggesting that student performance is apparently independent of pedagogical approaches [4] might be interpreted as implying the existence of a broad spectrum of individual differences in learning patterns. That is, some students learn most effectively in highly structured environments (lectures and demonstrations) while others find less structured situations (laboratories and self-study) more productive.

The simplest immediate solution to these rigidity problems is to provide parallel paths so that students can adapt their educational careers to those approaches which best suit their individual needs [3]. The computer-directed concept formation experiments developed by Project ITCH offer a new educational alternative to those students who find that the traditional approaches are not well suited to their individual needs.

Man-Machine Interaction

The man-machine approach to problem solving has been too long ignored in most academic departments even though it is already receiving considerable emphasis in modern industry. The inclusion of an interactive digital computer in the design process promises to revolutionize our approach to many engineering problems. An early introduction to the man-machine team approach to complex problems is an important addition to the undergraduate learning experience.

The ITCH Approach

The ITCH approach emphasizes the optimization of the man-machine interface in the creation of an effective learning environment. Whenever possible, we provide graphic I/O and natural language dialogs in the user's normal technical vocabulary and format. The prototype ITCH User station consists of a keyboard/printer for alpha-numeric I/O and a storage scope with joystick controlled cursor (with interrupt capability) for scope I/O. This combination allows experimentation with input and output on the scope or on the keyboard/printer or both.

In a typical student session problem parameters are specified by the student on a skeleton display provided by the specific problem package. The student uses the joystick controlled cursor (under the ADD command) [5] to specify geometric parameters (antenna locations, pole and zero locations, etc.). He uses the keyboard to specify numerical parameters (antenna excitations, semi-conductor doping levels, etc.). After the parameters are specified, the student may request display of results in a format suited to the problem and tailored to his interests (e.g., Bode or Nyquist formats for frequency domain information). The student may change geometric patterns (MOVE command) or numerical parameters (CHANGE command) and obtain the solutions with the new parameters [6]. He may save current results and then plot them over new results to gain insight into the effects of parameter changes (OVERLAY command). We believe that experimentation with problem parameters and the insights obtained by observing the effects of parameter changes are important steps in concept formation. The features outlined above are designed to maximize the convenience of man-machine interaction to encourage this experimentation.

As a student begins to understand parameter-process relations, it is possible to shift his attention toward design. He can then shift from, for example, studying how dipole excitations affect radiation patterns to designing dipole excitations to produce desired radiation patterns. In this way the student begins to develop an appreciation for the man-machine team approach to problem solving at an early point in his career. Also he is forced to develop insights into the effects of parameter changes or so that he is able to solve the design problems efficiently.

A primary constraint on the ITCH system is that it be low cost. We aim to attain costs per student terminal hour which are within the range of traditional university instructional costs. In a study of major engineering schools Terman found that these costs lie in the range of $2.50 to $5.00 per student hour [7]. Operational costs of the Phase I system have not been estimated since it is a demonstration project. A careful cost estimate for the proposed 16 terminal Phase II system shows that an operational student terminal hour cost of approximately $3.00 is realistic with current technology [8].

A second major constraint on the system is exportability--the ability to transfer software developed on Project ITCH to other schools. To achieve this we have avoided hardware modifications or the use of special hardware. All hardware in the Phase I and Phase II configuration is from the standard Digital Equipment Corporation PDP-8 line.

One final constraint on the Phase I software has been upward compatability with the proposed Phase II hardware. The TSS/8 Monitor used in Phase I must be modified for Phase II but the instructional packages developed in Phase I should be able to run on the Phase II system with negligible modification.

Current Project Status

Project ITCH is now halfway through Phase I. Phase I hardware has been installed and running for slightly more than one year, monitor modifications have been completed, and three instructional packages are fully operational.

Phase I Hardware

The Phase I hardware configuration is shown in Figure 1. This configuration is a minimum DEC PDP8/E time-sharing system with graphics capability added. Three keyboard/printer terminals [9] and one DEC modified Tektronix 611 storage scope terminal with joystick interrupt provide the input/output. This configuration is adequate for demonstrating all proposed system features and testing user response.

The DEC system was chosen primarily because at the time of purchase (June 1971) it was the only minicomputer time-sharing system offering terminal execution of assembly language programs.

Phase I Software

Phase I software development has involved two major projects: (a) the modification of the TSS/8 monitor, and (b) the development of instructional packages. Since the original TSS/8 monitor was not designed to handle scope I/O and had several other annoying limitations, an early Phase I project was to make monitor modification to:

(1) Reduce disk space required by resident monitor [10].
(2) Develop a scope interrupt handling procedure.
(3) Develop scope output interfaces.
(4) Develop scope input techniques.

The monitor modification is now complete. To ensure easy expandability for the Phase I software, all Phase I instructional packages run under the modified TSS/8 time-sharing monitor even though there is only one complete user station on the Phase I system.

The second major Phase I project is the development of Instructional Packages--simulation packages illustrating specific concepts. The basic structure of an instructional package is illustrated in Figure 2. The user interface provides natural language man-machine communication and certain housekeeping chores while the computational routines generate results from user input. Since the TSS/8 monitor limits each user to a 4K core memory space and the separate pieces in Figure 2a may require more than 4K of core, an automatic overlay system is included in each Instructional Package. Communication between overlays is provided through a reserved common block in core memory. (See Figure 2b).

A major problem in the programming of Instructional Packages has been the location of an appropriate high level programming language. Since all higher level languages running under the TSS/8 monitor are interpreted they do not provide convenient or appropriate vehicles for package development. Thus all programming has been done in assembly language. However, a library of subroutines for the user interfaces has been developed. Since much of the user interface function is a relatively simple symbol manipulation, the use of these library

subroutines has made the programming of the remainder of the package in assembly language reasonably convenient. On the other hand, programming computational routines in machine code is unnecessarily time consuming. The elimination of this problem will receive high priority in Phase II.

Phase II Hardware

Present plans for the Phase II hardware configuration are shown in Figure 3. This system is an expansion of the Phase I system--adding another processor, core memory, more disc and more terminals. The system would run under a modified version of the DEC TSS/8 monitor.

User Experience

User experience has been limited to demonstrations for EE faculty. During the winter and spring quarters of 1973, several third year EE classes will be using the system. Student experience will be reported during the verbal presentation of the paper.

Instructional Packages

Instructional Packages are developed to illustrate central concepts for the EE core courses which are difficult to teach by conventional methods. Many instructors find that certain concepts are not adequately conveyed in lectures and yet are not well suited to laboratory experiments. These difficult concepts which suggest potential Instructional Packages are identified by EE faculty members familiar with the ITCH approach. Following the identification of a problem area, a man-machine interaction scenario is developed to outline the structure of the package and the approach to be used in developing student insights. More consultation with the faculty is usually required to clarify and improve the scenario. The result of this iterative process is a detailed specification of the Instructional Package which is then programmed by a team of ITCH project programmers.

Three packages have been completed, another four are in progress, and three more have been identified for future development. The basic features of these ten packages are outlined below. The current status of these packages is shown in Figure 4.

1. Antenna Package

   Problem: What is the far field radiation pattern from a set of parallel dipole antennas with arbitrary locations (coplanar) and excitations?

   Input: Student uses joystick to locate dipoles in the plane of the scope face and types in the magnitude and phase of the excitation for each dipole.

   Output: Computer creates a polar plot of the relative field strength of the far field.

2. Bode/Nyquist Package

   Problem: What is the frequency response of a linear system? Will it be stable if feedback is added?

   Input: Student uses joystick to locate the poles and zeros of the system's transfer function in the s-plane displayed on the scope. The keyboard is used to specify the gain constant for the system.

   Output: Computer plots the frequency response of the system in either Bode format, Nyquist format, or on a linear frequency scale.

3. Point/Line Charge Package

   Problem: What is the static electric field produced by an array of point (or line) charges?

   Input: Student uses joystick to designate the locations of the charges on the scope face. He types in the magnitude of each charge as it is located.

Output: For each point the student designates with the joystick, computer plots the field line passing through that point.

4. Fourier Package

Problem: What is the relationship between the time and frequency domain representations of signals?

Input: Stude t uses joystick to specify a waveform on the scope face. He may also call a waveform from a library of waveforms and then modify it to suit his problem. He may specify filtering to be applied to the waveform. More complex options are also available.

Output: Computer plots the representation of the signal in the specified domain.

5. Static Field Package

Problem: Given a two dimensional arrangement of conducting surfaces and the potentials applied to each one, what is the resulting electrostatic field?

Input: Student uses joystick to specify the locations of the conducting surfaces, and an array of light buttons to specify the potential applied to each surface.

Output: For each point student designates with the joystick, computer draws the equipotential line passing through that point and labels it with its potential.

6. Transistor (BJT) Package

Problem: What are the relationships between the physical parameters of a bipolar junction transistor (BJT) and the carrier distributions and characteristics of the device?

Input: Student uses joystick to show base width on a picture of the device geometry. He specifies doping levels for the three regions through the keyboard.

Output: Computer plots the minority carrier distribution in the base region for specified voltages applied to the device. Characteristic curves for the device can be plotted.

7. Root Locus Package

Problem: A linear system is placed in a feedback loop with variable gain constant. How do the poles and zeros of the closed-loop system move as the gain constant is varied?

Input: Student uses joystick to locate the poles and zeros of the linear system in the s-plane displayed on the scope face. A linear scale on the display is used to specify the gain constant.

Output: For each student-specified gain constant, the computer plots the locations of the poles of the closed loop system.

8. Phase-Plane Package

Problem: What is the transient behavior of a second order linear or non-linear system?

Input: There are two cases: (1) student uses joystick to specify a linear or non-linear resistor in an RLC circuit, or (2) student types a second order linear or non-linear differential equation.

Output: Student uses joystick to specify a pair of initial conditions in the phase plane displayed on the scope. Computer plots the trajectory of the transient response beginning from the specified point.

9. Linear Filter Package

Problem: Given a two-port RLC filter, what are its impulse and frequency responses?

Input: The network topology is fixed. Student types in element types and parameters.

Output: Computer plots impulse and frequency responses for the specified filter.

10. Amplifier Design Package

Problem: The topology of a single stage amplifier is specified. How can the stages be interconnected and the component values chosen to meet overall amplifier specifications?

Input: Student specifies interconnection patterns and component values.

Output: Computer types operating point voltages and currents, and gain parameters. Frequency response of the amplifier can be plotted on request.

Some Typical Dialog

The flavor of a typical man-machine dialog is difficult to convey through words and pictures. However, in an attempt to go beyond the dry details presented above, we will describe two brief dialogs. While they lose much of their vitality when presented off-line, this appears to be the best that can be accomplished in the limited medium of print. To clarify steps in the interaction, the dialogs have been annotated and the student's TTY responses have been underlined. The * is the Instructional Package monitor ready symbol indicating that the monitor command recognizer is awaiting input. Thus it appears to the left of all monitor commands (ADD, MOVE, etc.).

The first dialog (Figures 5 and 6) is a conversation with the Antenna Package. The student is studying the radiation patterns of an array of two dipoles with various locations. Note that this package requires considerable switching of attention from scope to keyboard/printer. The student must enter all commands and numerical values through the keyboard. (Only the first letter of a command is required.)

The second dialog (Figures 7 and 8) is a conversation with the Point/Line Charge Package. The student is studying the static field created by two point end line charges. In this package we use light buttons for user commands to minimize the required attention to the keyboard/printer. Thus most of the printer output is strictly machine generated comments for introduction and record purposes. Once the user becomes familiar with the package he can ignore the printer and direct his attention entirely to the scope.

Conclusion

After one year of development Project ITCH has generated a number of Instructional Packages demonstrating its philosophy of interactive simulation to augment traditional instruction in the teaching of difficult engineering concepts. The effectiveness of the system will be strongly influenced by our Instructional Package design and careful attention to optimization of man-machine interaction. We expect that the next six months of user experience will provide a great deal of feedback for further improvement in both of these areas.

ACKNOWLEDGEMENT

The authors would like to acknowledge the assistance of the Project ITCH programming staff in the development of the software packages. The programmers contributing to this project are R. J. Davini, R. L. Domrese, R. W. Foss, P. S. Johannesson, D. Nassimi, R. A. Peterson, B. E. Pulk, and R. M. Salter. We are also indebted to our colleagues at the University of Minnesota for their many suggestions, comments, and criticisms.

NOTES AND REFERENCES

1. Concept formation in "learning through simulation" has been strongly advocated by analog computer users during the last 10 years. However, the analog computer approach was always seriously limited by problems of providing (a) adequate access to the computer, and (b) communication with the computer in a natural language. While the success of these analog experiments has demonstrated the value of interactive simulation [e.g., see various publications of the ACEUG (Analog Computer Education Users Group) and the journal Simulation], it is only the availability of reasonably priced interactive digital computer systems which finally makes this approach practical.

2. J. Holt, The Underachieving School. New York: Pitman, 1969.

3. M. Cohen, Deadline 79. Glynco Press, 1969.

4. R. Dubin and C. Teveggia, The Teaching-Learning Paradox. Center for the Advanced Study of Educational Administration, University of Oregon, 1968.

5. Comments in parentheses describe system commands (ADD, MOVE, etc.) which are illustrated in later examples (see the section, Some Typical Dialog)

6. A list of current values of the parameters is available at user's request (LIST command).

7. F. E. Terman, Economic factors relating to engineering problems. Journal of Engineering Education, 1969, 59(6), 510-514.

8. This figure includes operation and maintenance costs for computer, terminals, and software.

9. One keyboard/printer terminal is a DECwriter 30 cps terminal which offers character outputs at speeds closer to normal user reading rates.

10. This was necessary only because we have a small disk system (64 K words in the system).

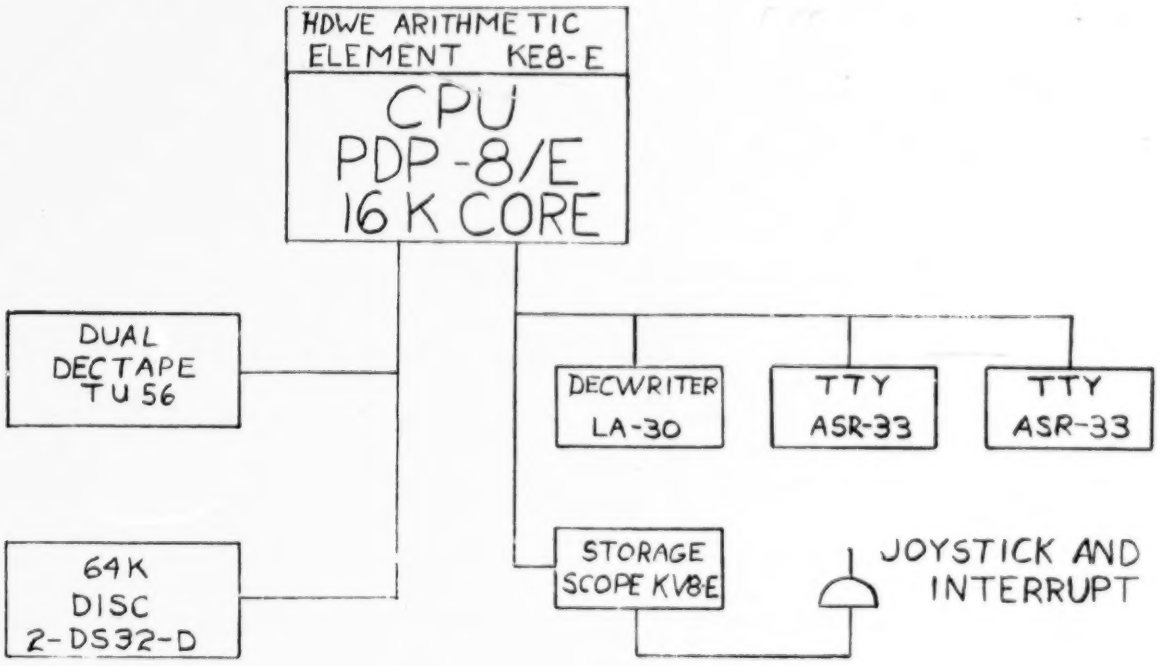
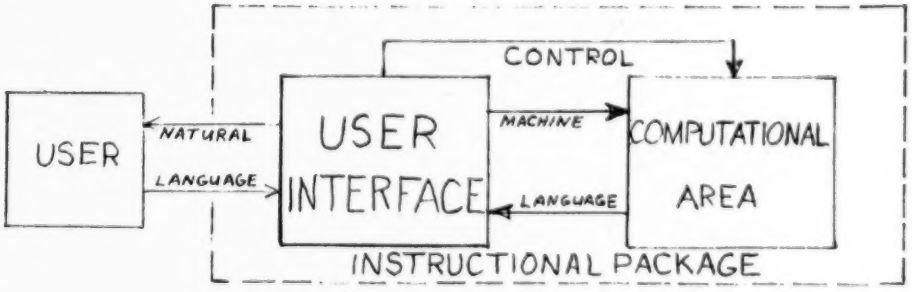
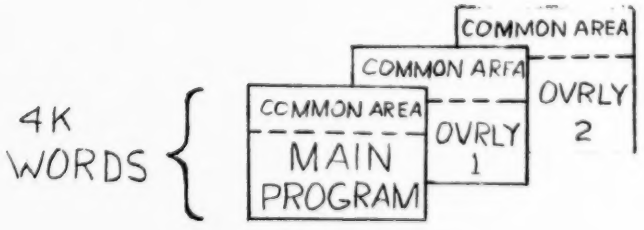


Figure 1
Phase I Hardware Configuration



a) Instructional Package Structure



b) Overlay Structure

Figure 2
Instructional Package Organization

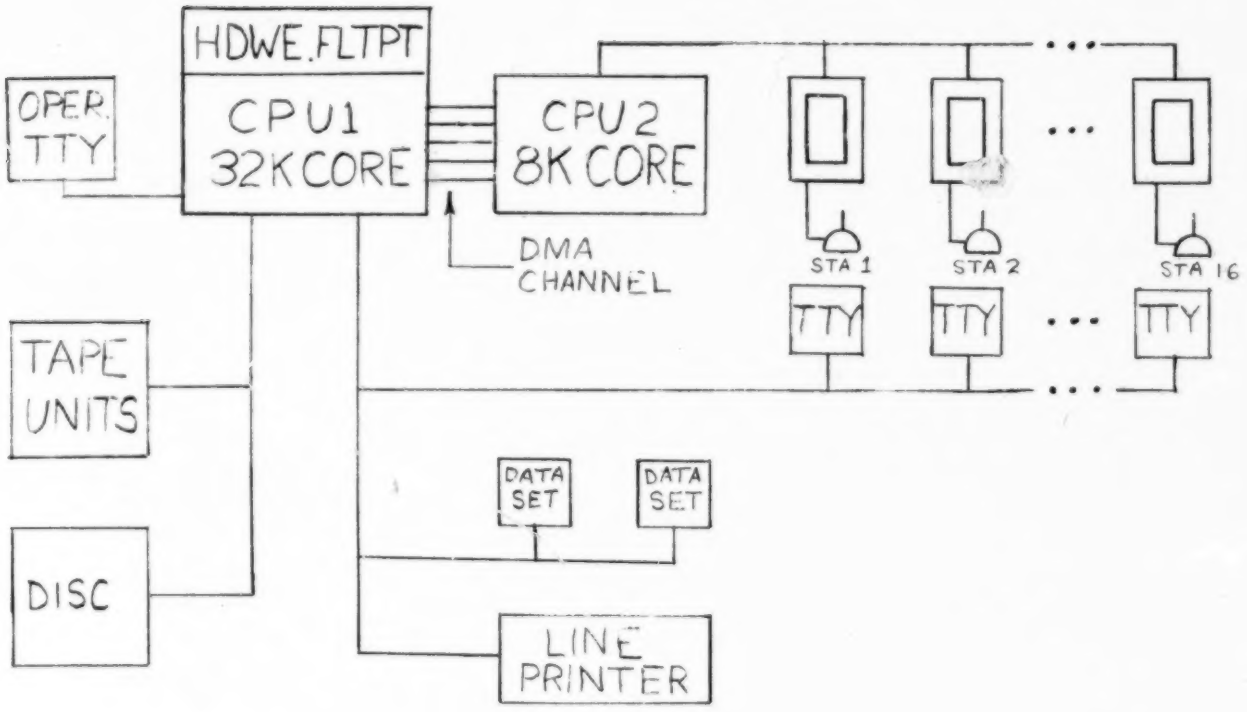


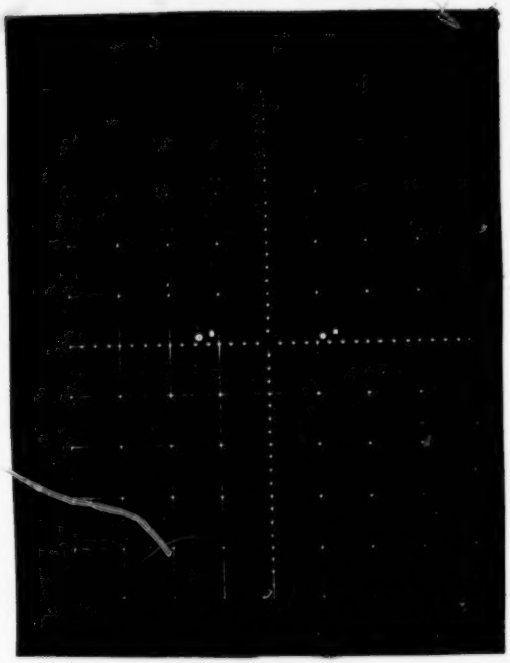
Figure 3
Phase II Hardware Configuration (Tentative)

	Status	Approxm. Man-Months Expended
Antenna Package	Completed	6
Bode/Nyquist Package	Completed	5
Point/Line Charge Package	Completed	4
Fourier Package	In Progress	3
Static Field Package	In Progress	.5
BJT Package	In Progress	.5
Root Locus Package	In Progress	1
Phase Plane Package	Proposed	0
Linear Filter Package	Proposed	0
Amplifier Design Package	Proposed	0

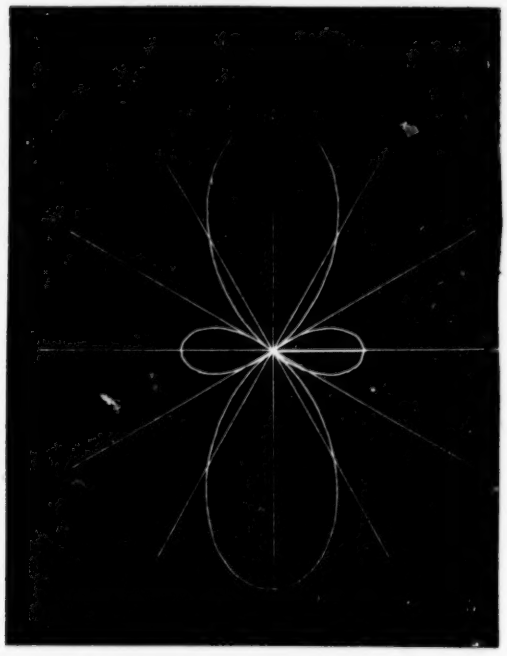
Figure 4
Instructional Package Status

```
•RUN ANTTNA                                          COMMENTS
ANTENNA PACKAGE                                      Call to Package
VERSIØN 2, . . . DECEMBER, 1972.

*ADD                                                 ADD Command

EACH LARGE DIVISIØN IS X WAVELENGTHS. X= .25

INPUT WITH JØYSTICK.

ELEMENT A:
AMPLITUDE = 1.
PHASE = 0.


ELEMENT B:
AMPLITUDE = 1.
PHASE = 0.
                                                     Photo 6a

*RUN                                                 RUN Command

FULL SCALE MAGNITUDE=+0.2000000E+01                  Photo 6b
DØ YØU WISH TØ SAVE CURRENT PATTERN?YES

*LIST                                                LIST Command

-------
+      .250 WAVELENGTHS/DIV.                         LIST Output

      X (WVLNTHS)     Y (WVLNTHS)     AMPLITUDE      PHASE (DEG.)

 A    +      .291     +      .029     +     1.000    +      .000
 B    -      .320     +      .037     +     1.000    +      .000
-------

*MØVE                                                MOVE Command

INDICATE ELEMENT TØ BE MØVED WITH JØYSTICK.
DELETED

INDICATE NEW PØSITIØN WITH JØYSTICK.
MØVE CØMPLETE.                                       Photo 6c

*RUN                                                 RUN Command

FULL SCALE MAGNITUDE=+0.2000000E+01
DØ YØU WISH TØ SAVE CURRENT PATTERN?NØ

*ØVERLAY                                             OVERLAY Command
                                                     Photo 6d
*EXIT                                                EXIT Command

DØ YØU WISH ME TØ SAVE DATA? NØ                      Student leaves package
↑BS
•
```

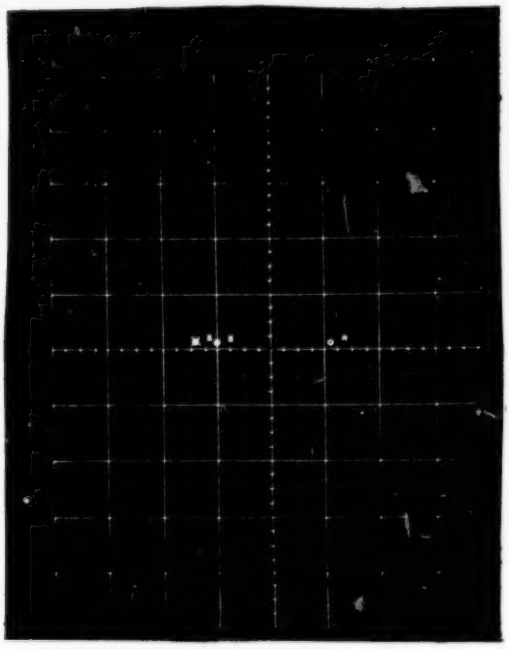
Figure 5
TTY Interaction with Antenna Package in Generating Radiation Patterns Shown in Figure 6



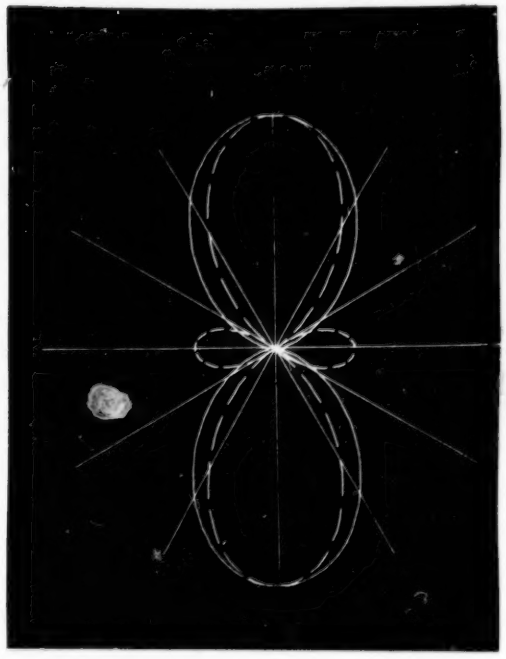
a) Student Locates Dipoles with Joystick



b) Machine Plots Requested Radiation Pattern



c) Student Moves Dipole B

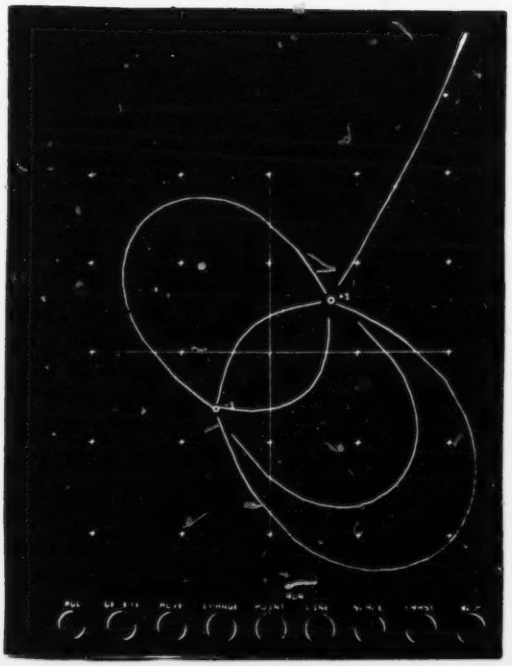


d) At Student's Request Machine Plots New Radiation Pattern and Overlays Previous One

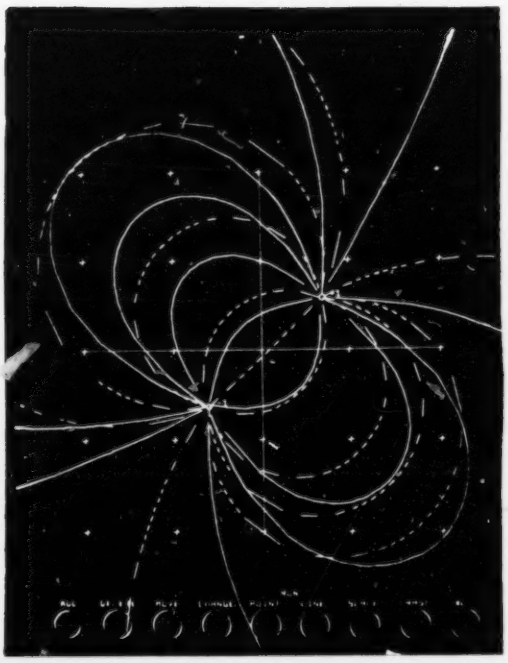
Figure 6
Antenna Package Scope Photographs

```
.RUN FIELD                                                    COMMENTS
FIELD DISPLAY.............VERSION1.....12-28-72               (L.B. = light button)
DOC FILE 35                                                   Call to package
PROGRAMERS:....R. A. PETERSON ....R. L. DOMRESE

ENTER COMMANDS WITH JOYSTICK

*ADD: INPUT CHARGES WITH JOYSTICK                             Student used ADD L.B.

Q= -1
                                                              Student inputs charge
Q= 1                                                          values

*POINT CHARGE FIELD PATTERN: START FIELD LINES WITH JOYSTICK   L.B.

*LINE CHARGE FIELD PATTERN: START FIELD LINES WITH JOYSTICK    L.B.

*ERASE:                                                       L.B.

*POINT CHARGE FIELD PATTERN: START FIELD LINES WITH JOYSTICK   L.B.

*CHANGE: SPECIFY CHARGES WITH JOYSTICK                        L.B.

Q= 2                                                          Student inputs new
                                                              value
*POINT CHARGE FIELD PATTERN: START FIELD LINES WITH JOYSTICK   L.B.

*CHANGE: SPECIFY CHARGES WITH JOYSTICK                        L.B.

Q= 1                                                          Student inputs new
                                                              value
*POINT CHARGE FIELD PATTERN: START FIELD LINES WITH JOYSTICK   L.B.

*CHANGE: SPECIFY CHARGES WITH JOYSTICK                        L.B.

Q= 4                                                          Student inputs new
                                                              value
*POINT CHARGE FIELD PATTERN: START FIELD LINES WITH JOYSTICK   L.B.

*QUIT:DO YOU WANT TO SAVE THE DATA BUFFER?NO                   Student leaves package
                                                              with QUIT L.B.

↑BS
.
```

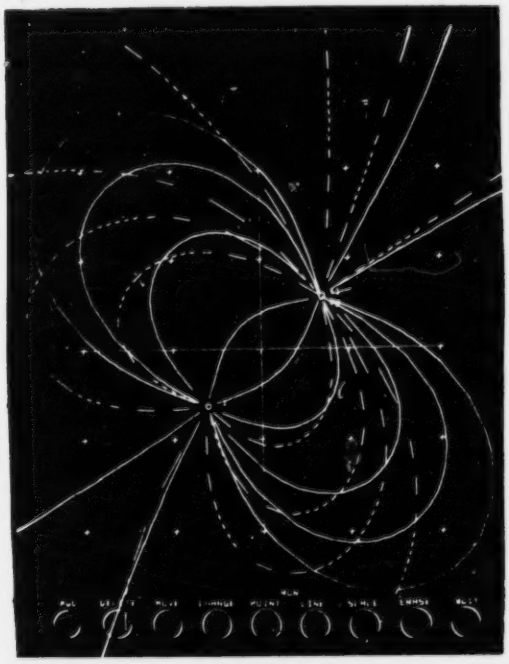
Figure 7
TTY Interaction with Point/Line Charge Package
in Generating Field Plots Shown in Figure 8



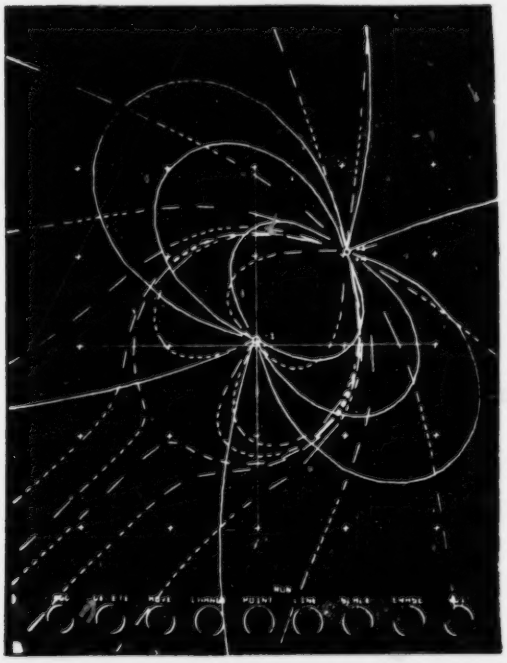
a) Student Locates Cursor with Joystick/Cursor and Plots a Few Lines



b) Student Plots More Field Lines for Point Charges and Overlays Field Lines for Line Charges



c) Student Overlays Plots for -1,1 Pair and -1,2 Pair of Point Charges



d) Student Overlays Plots for -1,1 Pair and -1,4 Pair of Point Charges

Figure 8
Point/Line Charge Scope Photographs